PATENT APPLICATION
Docket No. GE-0002US

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:		))
	Murren et al.	))
Serial No:	09/845,752	) Appeal No.)
Confirmation No:	3457	))
Filed:	4/30/2001	))
For:	Architecture and Process for Presenting Application Content to Clients	))))
Examiner	Siddiqi, M.	)

The Honorable Commissioner of Patents
Mail Stop Appeal Brief - Patents
P.O. BOX 1450
Alexandria, VA 22313-1450

BRIEF OF APPELLANT

The Applicant has filed a timely Notice of Appeal from the action of the Examiner in finally rejecting all of the claims that were considered in this application. This Brief is being filed under the provisions of 37 CFR 41.37. The Filing Fee, as set forth in 37 C.F.R. § 1.17(c), is submitted herewith.

TABLE OF CONTENTS

REAL PARTY IN INTEREST

The real party in interest is General Electric Capital Corporation, by way of assignment from Murren et al., who is the named inventive entity and is captioned in the present Brief.

RELATED APPEALS AND INTERFERENCES

None.

STATUS OF CLAIMS

Allowed Claims: No claims have been allowed.

Cancelled Claims: Claim 9.

Pending Claims: Claims 1-8 and 10-34 are pending in the application and stand finally rejected by the Examiner.

Appealed claims: Claims 1-8 and 10-34 form the basis for this appeal.

STATUS OF AMENDMENTS

No amendment has been made since the Final Action.

SUMMARY OF THE CLAIMED SUBJECT MATTER

The Application describes a multi-layer software architecture for efficient construction of server software applications including a presentation layer separate from a logic layer. The instant application generally describes a presentation layer divided into two tiers which structures presentation of responses. *Application, Abstract.*

Following is a brief summary of independent Claims 1, 12, 18, 24, 27 and 33 with exemplary references to the disclosure inserted for convenience. Specifically discussed dependent claims are also included in this list. References should not be understood as limiting any feature to the recited portions of the disclosure.

Claim 1 recites a server system, comprising:

one or more computers (112(1), 112(2), 112(3), 112(s)); and

an application (FIG. 1, 110) executing on the computers to handle client requests, the application comprising:

a business logic layer (FIG. 2, 204) to process (page 10, lines 14-28) the client requests according to a particular business domain and produce replies to be returned to the clients (FIG. 1, 102) in response to the client requests;

a presentation layer (FIG. 2, 212) separate from, but in communication with, the business logic layer to structure (page 13, lines 14-25) the replies in a manner that makes the replies presentable on different types of client devices according to a tag library containing pre-constructed tags (page 34, lines 19-28; FIG. 7, 702 for a variety of data formats; and

a request dispatcher (FIG. 2, 224) to structure a reply for service back to a client device, the request dispatcher being configured to access (page 34, lines 19-28) the tag library to obtain tags to structure the reply according to a particular data format.

Claim 2 recites a server system as recited in claim 1, wherein the application (FIG. 1, 110) is reconfigurable to other business domains by substituting other business logic layers that are designed to process the client requests according to the other business domains (page 7, lines 14-17).

Claim 12 recites in a server application that receives client requests for a problem domain and has at least one problem solving module to generate replies to be served back to clients, a presentation module separate from the problem solving module, comprising:

a presentation component (FIG. 7, 212) to construct how a reply will appear through use of a tag library (page 34, lines 19-28) (FIG. 7, 202) containing pre-constructed tags for a variety of data formats; and

a rendering component (260) to configure how the reply is output on a particular client (page 14, lines 16-22; page 34, lines 19-28).

Claim 13 recites a presentation module as recited in claim 12, wherein the presentation component (FIG. 7, 212) is configured to determine a layout of content to be included in the reply (page 14, lines 23-28).

Claim 14 recites a presentation module as recited in claim 12, wherein the presentation component (FIG. 7, 212) is configured to determine display attributes for the reply (page 35, lines 1-11).

Claim 18 recites a computer software architecture embodied on one or more computer-readable media, comprising:

a presentation tier (FIG. 2, 212) to determine how data for communication to a client device is to be presented on the client device through use of a tag library containing pre-constructed tags for a variety of data formats (page 34, lines 19-28) (FIG. 7, 202); and

a rendering tier (260), separate from the presentation tier, to determine how to render the data on the client device (page 14, lines 16-22; page 34, lines 19-28).

Claim 19 recites a computer software architecture as recited in claim 18, wherein the presentation tier is configured to determine at least one of (1) a layout of the data, (2) a color scheme in which to present the data, (3) a presentation theme, and (4) a particular skin appearance (page 33, lines 13-17).

Claim 23 recites a computer software architecture as recited in claim 18, wherein the presentation tier comprises:

a tag library (page 34, lines 19-28) (FIG. 7, 202)containing pre-constructed tags for a variety of data formats; and

a request dispatcher (FIG. 2, 224) to structure the data using the tags from the tag library, the tags being selected to structure the data in a manner that is supported by the client device (page 34, lines 19-28).

Claim 24 recites an architecture comprising:

a tag library (page 34, lines 19-28) (FIG. 7, 202) containing pre-constructed tags for a variety of data formats;

multiple request dispatchers (page 14, line 4) (FIG. 7, 202) to structure replies to be returned to client devices in response to requests submitted by the client devices, individual request dispatcher formatting data according to particular formats that are supported by the client devices (page 14, lines 6-11) according to the tag library; and

content renderer to conform the replies to output capabilities of the client devices to which the replies are to be returned (page 14, lines 16-22).

Claim 27 recites a method comprising:

receiving (302) a reply generated by a server application in response to a client request;

structuring (316, 800) the reply to define how the reply will appear when communicated to and presented at the client through use of a tag library (page 36, lines 6-28)(page 34, lines 19-28) (FIG. 7, 202) containing pre-constructed tags for a variety of data formats (page 36, lines 20-28); and

independent of said structuring, conforming (808) the reply to output capabilities of the client (page 37, lines 1-6).

Claim 33 recites one or more computer-readable media comprising computer-executable instructions that, when executed, direct an application server to:

generate (page 9, line 20) replies in response to client requests through use of a tag library (page 34, lines 19-28) (FIG. 7, 202) containing pre-constructed tags for a variety of data formats, the client requests being submitted by diverse client devices that support different data formats and different communication protocols (page 37, lines 1-6); and

structure (808) the replies to define how the replies will appear when communicated to and presented on the client devices and independently form individual replies for output capabilities of the client devices so that the replies are encoded to comply with the data formats supported by the client devices and are sent using the communication protocols of the client devices (page 36, lines 20-28).

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Claims 1-8 and 10-34 stand rejected 35 U.S.C. §102(e) over United States Patent 6,643,652 to Helgeson et al. (Hereinafter, "Helgeson")

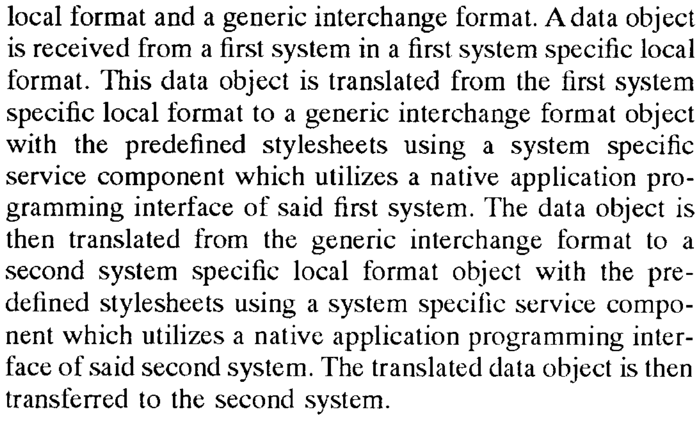
ARGUMENT

4. **FIRST GROUND OF REJECTION:**

Claims 1-8 and 10-34 satisfy the requirements of 35 U.S.C. § 102(e) and therefore are not anticipated by Helgeson. For clarity, Applicants will make reference to the claim and claim language most nearly associated with the Response to Arguments section of the Final Action, as the this section of the Final Action does not reference specific claims.

i. Helgeson discloses a common message passing system which fails to disclose a "business logic layer" as recited in the claims (in particular **Claim 1**).

The Helgeson reference discloses a data exchange system for passing data among systems on a network. *Helgeson, Abstract.* For example, Helgeson describes managing data exchange by using predefined stylesheets which map the exchange between a system specific format and a generic format. In this way, a data object in a first system format (1) may be translated into the generic format (2) for subsequent translation into a second system specific format (3) in accordance with the stylesheet. Thus, the stylesheet defines the entire path for the translation between the formats. *Helgeson, Col. 2, lines 51-67.* The Helgeson reference does this in order to allow data communication across multiple system platforms using different formats. For example, Helgeson describes the following transfer of data,



local format and a generic interchange format. A data object is received from a first system in a first system specific local format. This data object is translated from the first system specific local format to a generic interchange format object with the predefined stylesheets using a system specific service component which utilizes a native application programming interface of said first system. The data object is then translated from the generic interchange format to a second system specific local format object with the predefined stylesheets using a system specific service component which utilizes a native application programming interface of said second system. The translated data object is then transferred to the second system.

Helgeson, Col. 2, lines 55-67.

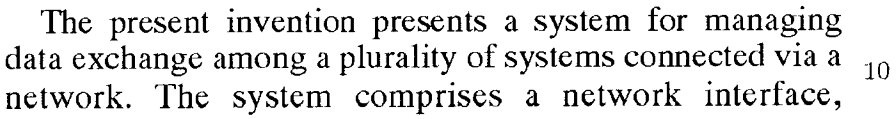
In contrast, Claim 1, in part recites,

- “a business logic layer to process the client requests according to a particular business domain and produce replies to be returned to the clients in response to the client requests;”

Helgeson does not process the client requests because data is not processed to produce replies to the client according to a particular business domain. In Helgeson, the ultimate result, and the goal of the patent, is that the data object is in the second format (on the second system) rather than being processed for return to the client. This is to say, processing and returning a reply to the client is contrary to Helgeson because Helgeson is attempting to get the data from a first system, implement a first format, to a second system, (in a format understandable to the second system) in order “manage data exchange” (rather than “processing” as contended by the

Examiner). *Helgeson, Abstract.* Thus, in Helgeson, there is no reply because the end goal is to manage the transfer of data objects in a network rather than solving problems.

Applicant disagrees with the Advisory Action which contends that Applicant is narrowly reading 1) the reference and 2) the claim language. This contention is incorrect because it ignores the language of Claim 1 which in part recites "producing replies to be returned to the clients". The foregoing is also contrary to Helgeson because this would not be "management" of data objects being passed around (i.e., received or sent). This "narrow" contention does not take into account the numerous teaching in Helgeson which, in essence, repeats the following:



> The present invention presents a system for managing data exchange among a plurality of systems connected via a network. The system comprises a network interface,

Helgeson, Col. 3, lines 9-11.

In Helgeson, the purpose is to permit integration of disparate applications which may have data in many different formats and allow data to be passed between systems which may be incompatible with each other. *Helgeson, Col. 1, lines 34-40.* While the Examiner is obligated to take a "broad interpretation" of claim terms under *M.P.E.P. §2111,* the "broad interpretation" of one or more terms cannot put the term at odds with other terms within the claim. Thus, a "broad interpretation" cannot render the claim language internally inconsistent with the term's usage in the claim.

In contrast, Claim 1 recites that the business logic layer "processing the client request according to a particular business domain and produces replies to be returned to the clients." In this way, processing requests are processed according to the business domain and then returned to the clients. For example, the business logic layer produces solutions for a problem domain. *Application, Page 10, lines 15-17.* This is not a "narrow" reading because nowhere does Helgeson disclose that the purpose of the subject matter is to produce solutions rather than manage data transfers. Even if we presume, for the sake of argument, the contention in the Final Action is correct, the Action has failed to cite a portion of Helgeson which discloses the Examiners assertion with at least as much clarity as the portion above supports the Applicant's position. The Examiner's contention by contrast, seems to ignore the recitation of producing replies.

Applicant disagrees with the contention that the "common business objects layer" discloses the business logic layer. The contention is incorrect because the common business objects layer in Helgeson is merely a set of common business objects which are shared across the SABA applications. For example, business objects include party, plan or calendar. Helgeson, Col. 8, lines 4-8. As a result, the common business objects do not process client request but instead provide common data such as organization membership, schedules, employees, clients, departments and so on. *Helgeson, Col. 8, lines 4-30.*

ii. The Helgeson "stylesheets" fail to disclose a tag library.

The contention that "stylesheets" are equivalent to tags in a tag library is incorrect. First, Claim 1, in part, recites,

> - "a presentation layer separate from, but in communication with, the business logic layer to structure the replies in a manner that makes the replies presentable on different types of client devices according to a tag library containing pre-constructed tags for a variety of data formats; and
> - a request dispatcher to structure a reply for service back to a client device, the request dispatcher being configured to access the tag library to obtain tags to structure the reply according to a particular data format."

In Helgeson, the stylesheets are akin to a driving direction map. Say for example, you want to get from city "A" to city "B" (in which the cities are akin to the first and second systems) the stylesheet would lay out a (single) path for you to drive such as if a person took a highlighter and noted the highway you are to travel on. In Helgeson, the stylesheet is a single translation path between a system specific local format and a generic format.

The citation of Helgeson, Col. 51, line 65-Col. 52, line 2, fails to anticipate a request dispatcher as recited in Claim 1. The Helgeson passage does not disclose the recited features as Helgeson points out that the stylesheets define and control the presentation of the output to the user. *Helgeson, Col. 50, lines 9-11.* Additionally, the passage merely describes "pre-process model file" (FIG. 8B), thus the passage does not disclose "obtain tags to structure the reply according to a particular data format" as is recited.

In contrast, tags within a tag library, as recited, may be more akin to having multiple discrete directions, thus while you know you are going from east to west from city "A" to city "B" you have the possibility of taking many different routes which are split-up into many individual directions (e.g., take a left and follow and alternate route, you can take Lincoln Way or First Street). In this way, the tag library may be accessed "to obtain tag to structure the reply according to a particular data format" rather than merely following a single set direction laid out in a stylesheet. Support for this position may at least be found in the following passage,

> The request dispatcher type 264 may optionally access a tag library 702 to retrieve preformed HTML tags for convenient and efficient construction of a response page. The tag library 702 contains various tags that adapt to various device formats and protocols, as well as different languages. In this way, a single code base can be used to ensure a correct format for multiple languages, to selectively render output based on the type of device that is receiving the output of the application, and so forth. . *Application, Page 34, lines 19-26.*

In Helgeson, each translation requires a map. Correspondingly, an entire map is necessary for each translation and only one result occurs per translation. In contrast, the utilization of a tag library making use of pre-constructed tags, may permit adaptation to various device formats and protocols. For example, a program tag includes a name attribute that specifies the name of the program. *Instant Application*, Page 27, lines 4-5. Other tag attributes are also possible. Other tags and tag attributes permit adaptation as discussed in the Application. Reversal of the pending rejection is respectfully requested.

iii. **Claim 2** meets the requirements under 35 U.S.C. §102(e) due to the claim's dependence from Claim 1 which meets the requirements of §102. Claim 2 is additionally allowable because Helgeson fails to disclose an application being reconfigurable.

The contention that Helgeson discloses application reconfiguration to other business domains is incorrect. Not only does Helgeson fail to teach a reconfiguration for other domains, but this contended disclosure would conflict with the express purpose of utilizing a single generic transfer language (e.g., all local formats are translated into a common format). For example, in Helgeson, changing business domains would necessitate changing <u>each and every</u> stylesheet to the new domain to ensure proper translation. Helgeson attempts to avoid this translation problem by translating specific local formats into a (single) generic format for passage. Reconfiguration of the Helgeson system would in essence require a "ground-up" rebuilding of the Helgeson system. Reversal of the pending rejection is respectfully requested.

iv. **Claims 2-8, 10 and 11** depend either directly or indirectly from Claim 1 and are allowable as depending from an allowable base claim, as well as for their own recited features. Each of these claims depends from Claim 1 and is therefore allowable for reasons discussed with respect to Claim 1. These claims are also allowable for their own recited features which, in combination with those recited in Claim 1, are not disclosed in the references of record. Reversal of the pending rejection under 35 U.S.C. §102(e) to Claims 2-8, 10 and 11 is requested.

v. Helgeson fails to teach a tag library containing pre-constructed tags. As such, **Claim 12** meets the conditions of 35 U.S.C. §102(e).

> Independent Claim 12, in part, recites,
> • "a presentation component to construct how a reply will appear through use of a tag library containing pre-constructed tags for a variety of data formats; and
> • a rendering component to configure how the reply is output on a particular client."

Helgeson discloses the use of stylesheets as discussed above with respect to Claim 1. Applicant notes that contrary to the contention in the

Final Action, Claims 1 and 12 contain different language thus, the pending rejection of Claim 12 does not address all the claim terminology.

In particular, Helgeson does not disclose a "presentation component to construct how a reply will appear through use of a tag library containing pre-constructed tags." Helgeson does not implement a tag library or "configure how the reply is output on a particular client" as the Helgeson stylesheets define the data format and only a single stylesheet is available for each translation. Thus, configuration is not necessary in Helgeson because there can be only one interpretation (as defined by the stylesheet) for each translation. Reversal of the pending rejection under 35 U.S.C. §102(e) to Claim 12 is requested.

vi. **Claims 13 and 14** meet the requirements of 35 U.S.C. §102(e) as Helgeson fails to "determine a layout of content" (Claim 13) and "determine display attributes for the reply"(Claim 14).

Consistent with the Arguments in Claim 12 (above), in the Helgeson reference, there is no need to determine the layout of content because the stylesheet "maps" the entire translation, as a result, the layout does not have to be determined because using but one map per translation would result in but one consistent layout as the map controls the layout of the data object. Similarly, determining "display attributes for the reply" is equally unnecessary in Helgeson as the stylesheets provide a uniform translation.

vii. **Claims 13-17** depend from Claim 12 and are allowable as depending from an allowable base claim, as well as for their own recited features.

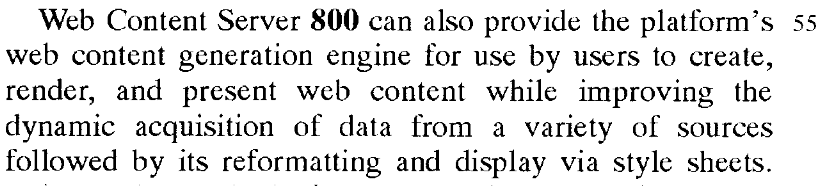
Each of these claims depends from Claim 12 and is therefore allowable for reasons discussed with respect to Claim 12. These claims are also allowable for their own recited features which, in combination with those recited in Claim 1, are not disclosed in the references of record. Reversal of the pending rejection under 35 U.S.C. §102(e) to Claims 13-17 is requested.

viii. Independent **Claim 18** meets the requirements of 35 U.S.C. §102(e) and is not anticipated by Helgeson.

Applicant notes that contrary to the contention in the Final Action, Claims 1 and 18 contain different language thus, the pending rejection of Claim 18 does not address all the claim terminology. Claim 18, in part recites,

> - "a presentation tier to determine how data for communication to a client device is to be presented on the client device through use of a tag library containing pre-constructed tags for a variety of data formats; and
> - a rendering tier, separate from the presentation tier, to determine how to render the data on the client device."

Helgeson fails to disclose at least these features, as generally discussed above. Helgeson doe not anticipate Claim 18 because Helgeson does not implement a tag library. Helgeson uses stylesheets which entirely define the translation between different formats (and presentation), as a result, the use of tags which are pre-constructed for a variety of data formats, is inconsistent with Helgeson's stylesheets. While the Examiner contends that Helgeson, Col. 49, lines 19-21 (discussing a web content server) disclosed a presentation layer (Claim 1) (which Applicant presumes is being asserted with respect to Claim 18), this disclosure fails to indicate that presentation of the data is dictated by the stylesheet. In contrast, Helgeson, Col. 49, lines 55-59 supports the Applicant's position that Claim 18 is not disclosed by Helgeson because the stylesheets define the presentation, as a result, there is no need to determine how to render the data or to use a tag library as presentation is dictated by the stylesheets.

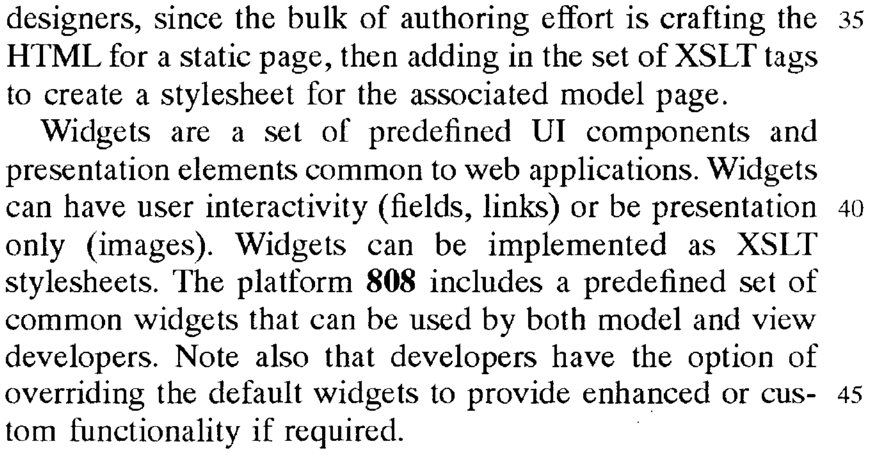


> Web Content Server **800** can also provide the platform's web content generation engine for use by users to create, render, and present web content while improving the dynamic acquisition of data from a variety of sources followed by its reformatting and display via style sheets.

Helgeson, Col. 49, lines 55-59.

For at least the foregoing reasons, reversal of the pending rejection under 35 U.S.C. §102(e) to Claim 18 is requested.

ix. **Claim 19** meets the requirements of 35 U.S.C. §102(e) as Helgeson fails to disclose "wherein the presentation tier is configured to determine at least one of (1) a layout of the data, (2) a color scheme in which to present the data, (3) a presentation theme, and (4) a particular skin appearance. In the Final Rejection, the Examiner cited Helgeson, Col. 51, lines 35-46 for this proposition. Applicant disagrees. The cited portion of Helgeson is provided below for the Board's convenience.



> designers, since the bulk of authoring effort is crafting the HTML for a static page, then adding in the set of XSLT tags to create a stylesheet for the associated model page.
>
> Widgets are a set of predefined UI components and presentation elements common to web applications. Widgets can have user interactivity (fields, links) or be presentation only (images). Widgets can be implemented as XSLT stylesheets. The platform **808** includes a predefined set of common widgets that can be used by both model and view developers. Note also that developers have the option of overriding the default widgets to provide enhanced or custom functionality if required.

Helgeson, Col. 51, lines 35-46.

As the Board is well aware, "[a]n anticipating reference must describe the patented subject matter *with sufficient clarity and detail* to establish that the subject matter existed and that its existence *was recognized by persons of ordinary skill in the field of invention.*" *ATD Corp.v. Lydall, Inc.,* 48 USPQ.2d 1321,1328 (Fed. Cir. 1998) citing *In re Spada*, 15 USPQ.2d 1655, 1657 (Fed. Cir. 1990). Emphasis added. Helgeson does not disclose these features. Thus, the Helgeson reference does not anticipate the present claim. Reversal of the pending rejection under 35 U.S.C. §102(e) to Claim 19 is requested

x. **Claims 20-22** depend directly from Claim 18 and are allowable as depending from an allowable base claim, as well as for their own recited features. Each of these claims depends from Claim 18 and is therefore allowable for reasons discussed with respect to Claim 18. These claims are also allowable for their own recited features which, in combination with those recited in Claim 1, are not disclosed in the references of record. Reversal of the pending rejection under 35 U.S.C. §102(e) to Claims 20-22 is requested.

xi. **Claim 23** meets the requirements of 35 U.S.C. §102(e) as Helgeson fails to at least disclose "a request dispatcher to structure the data using the tags from the tag library, the tags being selected to structure the data in a manner that is supported by the client device." Emphasis added. The Examiner improperly contends that the tags are "stylesheets". This contention is incorrect as there is no need to "select" because a single stylesheet dictates (via mapping) all aspects of the presentation. This is to say that Helgeson does not disclose "the tags being selected to structure the data in a manner that is supported by the client device" because there is no choice but to use the stylesheet for the particular translation which dictates the structure of the data. Reversal of the pending rejection under 35 U.S.C. §102(e) to Claim 23 is requested.

xii. Independent **Claim 24** meets the requirements of 35 U.S.C. §102(e) and is not anticipated by Helgeson.

Applicant notes that contrary to the contention in the Final Action, Claims 1 and 24 contain different language thus, the pending rejection of Claim 24 does not address all the claim terminology. Claim 24, in part, recites,

> • "a tag library containing pre-constructed tags for a variety of data formats;
> • multiple request dispatchers to structure replies to be returned to client devices in response to requests submitted by the client devices, individual request dispatcher formatting data according to particular formats that are supported by the client devices according to the tag library; and
> • content renderer to conform the replies to output capabilities of the client devices to which the replies are to be returned."

As discussed above, Helgeson fails to disclose a tag library for a variety of data formats. Additionally, Helgeson fails to teach 1) "multiple request dispatchers to structure replies to be returned to client devices" and 2) "content renderer to conform the replies to output capabilities of the client devices". Helgeson fails to at least teach items 1 and 2 as Helgeson does not "return . . .responses" but instead, manages communications between systems which are not returned. Moreover, in Helgeson there is no need to format "data according to particular formats that are supported by the client devices according to the tag library" as a stylesheet dictates the particular format. This is to say for a generic-specific translation the stylesheet defines the formatted outcome. In addition, Helgeson, Col. 51, lines 47-67 disclose that tags are not used to format end-content formation according to particular formats. As a result, widgets may be implemented in stylesheets in the transform step and aid in content generation. *Helgeson, Col. 51, lines 47-50.* Reversal of the pending rejection under 35 U.S.C. §102(e) to Claim 24 is requested.

xiii. **Claims 25-26** depend directly from Claim 24 and are allowable as depending from an allowable base claim, as well as for their own recited features. Each of these claims depends from Claim 24 and is therefore allowable for reasons discussed with respect to Claim 24. These claims are also allowable for their own recited features which, in combination with those recited in Claim 24, are not disclosed in the references of record. Reversal of the pending rejection under 35 U.S.C. §102(e) to Claims 25-26 is requested.

xiv. Independent **Claim 27** meets the requirements of 35 U.S.C. §102(e) and is not anticipated by Helgeson.

Applicant notes that contrary to the contention in the Final Action, Claims 1 and 24 contain different language than Claim 27 thus, the pending rejection of Claim 27 does not address all the claim terminology. Claim 27, in part, recites,

- "receiving a reply generated by a server application in response to a client request;
- structuring the reply to define how the reply will appear when communicated to and presented at the client through use of a tag library containing pre-constructed tags for a variety of data formats; and
- independent of said structuring, conforming the reply to output capabilities of the client."

Helgeson does not disclose at least these features. In particular, Helgeson does not "structure the reply" because Helgeson is directed to managing data object transfers rather than solving client requests. Even if we assume, for arguments sake, that Hegelson structures replies, Helgeson does not use a "tag library containing pre-constructed tags for a variety of data formats" because Helgeson discloses the use of stylesheets which mandate translation for a specific local/generic format. As a result, in Helgeson, the reply is not to output capabilities of the client because a common stylesheet would be used for all clients employing the same format. Reversal of the pending rejection under 35 U.S.C. §102(e) to Claim 27 is requested.

xv. **Claims 28-32** depend directly from Claim 27 and are allowable as depending from an allowable base claim, as well as for their own recited features. Each of these claims depends from Claim 27 and is therefore allowable for reasons discussed with respect to Claim 27. These claims are also allowable for their own recited features which, in combination with those recited in Claim 27, are not disclosed in the references of record. Reversal of the pending rejection under 35 U.S.C. §102(e) to Claims 28-32 is requested.

xvi. Independent **Claim 33** meets the requirements of 35 U.S.C. §102(e) and is not anticipated by Helgeson.

Applicant notes that contrary to the contention in the Final Action, Claims 1 and 8 contain different language than Claim 33 thus, the pending rejection of Claim 33 does not address all the claim terminology. Claim 33, in part, recites,

> - "generate replies in response to client requests through use of a tag library containing pre-constructed tags for a variety of data formats, the client requests being submitted by diverse client devices that support different data formats and different communication protocols; and
> - structure the replies to define how the replies will appear when communicated to and presented on the client devices and independently form individual replies for output capabilities of the client devices so that the replies are encoded to comply with the data formats supported by the client devices and are sent using the communication protocols of the client devices.

The Helgeson reference does not anticipate Claim 33 because the Examiner has failed to show how the Helgeson stylesheets (which control translation between local format to generic format) permit use with "client devices that support different data formats and different communication protocols". Additionally, Helgeson does not "generate replies in response to client requests" but instead manages data exchange among systems in a network. *Helgeson, Abstract.* Moreover, Helgeson does not "form individual replies for output capabilities of the client devices so that the replies are encoded to comply with the data formats supported by the client devices and are sent using the communication protocols of the client devices" because each translation uses a stylesheet which is not described as forming "individual replies for output capabilities of the client devices". Reversal of the pending rejection under 35 U.S.C. §102(e) to Claim 33 is requested.

xvii. **Claim 34** depends directly from Claim 33 and is allowable as depending from an allowable base claim, as well as for the claim's own recited features. Each of these claims depends from Claim 33 and is therefore allowable for reasons discussed with respect to Claim 33. These claims are also allowable for their own recited features which, in combination with those recited in Claim 33, are not disclosed in the references of record. Reversal of the pending rejection under 35 U.S.C. §102(e) to Claim 33 is requested.

Conclusion

The Applicant respectfully considers this application to be in condition for allowance and respectfully requests the Board to overturn the final rejection and that the Examiner pass this application to allowance.

Dated this 23rd day of July, 2007

Respectfully Submitted,

Nathan T. Grebasch
Attorney for the Applicant
Reg. No. 48,600

LEE & HAYES PLLC
421 W. Riverside Avenue,
Suite 500
Spokane WA, 99201
Telephone: (509) 324-9256 (228)
Fax: (509) 323-8979

CLAIMS APPENDIX

1. A server system, comprising:

one or more computers; and

an application executing on the computers to handle client requests, the application comprising:

a business logic layer to process the client requests according to a particular business domain and produce replies to be returned to the clients in response to the client requests;

a presentation layer separate from, but in communication with, the business logic layer to structure the replies in a manner that makes the replies presentable on different types of client devices according to a tag library containing pre-constructed tags for a variety of data formats; and

a request dispatcher to structure a reply for service back to a client device, the request dispatcher being configured to access the tag library to obtain tags to structure the reply according to a particular data format.

2. A server system as recited in claim 1, wherein the application is reconfigurable to other business domains by substituting other business logic layers that are designed to process the client requests according to the other business domains.

3. A server system as recited in claim 1, wherein the presentation layer is configured to determine a layout of content in the replies.

4. A server system as recited in claim 1, wherein the presentation layer is configured to determine display attributes in the replies.

5. A server system as recited in claim 1, wherein the different types of client devices support different data formats, the presentation layer being configured to select appropriate data formats for encoding the replies.

6. A server system as recited in claim 1, wherein the different types of client devices support different communication protocols, the presentation layer being configured to select appropriate communication protocols for delivering the replies to the clients.

7. A server system as recited in claim 1, wherein the presentation layer is configured to determine how to display the replies for a particular client.

8. A server system as recited in claim 1, wherein the presentation layer comprises:

a presentation tier to determine how the replies will appear on the client devices to users; and

a rendering tier, separate from the presentation tier, to determine how to render the replies on the client devices.

9. (cancelled).

10. A server system as recited in claim 1, wherein the request dispatcher is configured to select a communication protocol to be used to serve the reply back to the client device.

11. A server system as recited in claim 1, wherein the presentation layer further comprises a content renderer to conform the reply structured by the request dispatcher to output capabilities of the client device to which the reply will be returned.

12. In a server application that receives client requests for a problem domain and has at least one problem solving module to generate replies to be served back to clients, a presentation module separate from the problem solving module, comprising:

a presentation component to construct how a reply will appear through use of a tag library containing pre-constructed tags for a variety of data formats; and

a rendering component to configure how the reply is output on a particular client.

13. A presentation module as recited in claim 12, wherein the presentation component is configured to determine a layout of content to be included in the reply.

14. A presentation module as recited in claim 12, wherein the presentation component is configured to determine display attributes for the reply.

15. A presentation module as recited in claim 12, wherein the clients support different data formats, the presentation component being configured to select an appropriate data format for encoding the reply for the particular client.

16. A presentation module as recited in claim 12, wherein the clients support different communication protocols, the presentation component being configured to select an appropriate communication protocol for delivering the reply to the particular client.

17. A presentation module as recited in claim 12, wherein the rendering component is configured to conform the reply to a specific display at the particular client.

18. A computer software architecture embodied on one or more computer-readable media, comprising:

a presentation tier to determine how data for communication to a client device is to be presented on the client device through use of a tag library containing pre-constructed tags for a variety of data formats; and

a rendering tier, separate from the presentation tier, to determine how to render the data on the client device.

19. A computer software architecture as recited in claim 18, wherein the presentation tier is configured to determine at least one of (1) a layout of the data, (2) a color scheme in which to present the data, (3) a presentation theme, and (4) a particular skin appearance.

20. A computer software architecture as recited in claim 18, wherein the presentation tier is configured to select a data encoding format for encoding the data and a communications protocol in which to send the data to the client device.

21. A computer software architecture as recited in claim 18, wherein the presentation tier comprises multiple dispatchers, each dispatcher being configured to encode the data according to a particular encoding format.

22. A computer software architecture as recited in claim 18, wherein the presentation tier comprises multiple dispatchers, each dispatcher being configured to package the data according to a particular communications protocol.

23. A computer software architecture as recited in claim 18, wherein the presentation tier comprises:

a tag library containing pre-constructed tags for a variety of data formats; and

a request dispatcher to structure the data using the tags from the tag library, the tags being selected to structure the data in a manner that is supported by the client device.

24. An architecture comprising:

a tag library containing pre-constructed tags for a variety of data formats;

multiple request dispatchers to structure replies to be returned to client devices in response to requests submitted by the client devices, individual request dispatcher formatting data according to particular formats that are supported by the client devices according to the tag library; and

content renderer to conform the replies to output capabilities of the client devices to which the replies are to be returned.

25. An architecture as recited in claim 24, wherein individual request dispatchers are further configured to select communication protocols to be used to serve the replies back to the client devices.

26. An architecture as recited in claim 24, wherein the content renderer is configured to conform the replies to specific display types at the client devices.

27. A method comprising:

receiving a reply generated by a server application in response to a client request;

structuring the reply to define how the reply will appear when communicated to and presented at the client through use of a tag library containing pre-constructed tags for a variety of data formats; and

independent of said structuring, conforming the reply to output capabilities of the client.

28. A method as recited in claim 27, wherein the structuring comprises selecting an encoding format in which to encode the reply.

29. A method as recited in claim 27, wherein the structuring comprises selecting a communication protocol for sending the reply to the client.

30. A method as recited in claim 27, wherein the structuring comprises selecting at least one of (1) a layout of content in the reply, (2) a color scheme of the reply, (3) a skin theme, and (4) a logo to brand the reply.

31. A method as recited in claim 27, further comprising:

storing pre-constructed tags that can be used to construct the reply in different formats; and

selecting at least one of the tags when structuring the reply.

32. A method as recited in claim 27, wherein the configuring comprises sizing the reply for a display at the client.

33. One or more computer-readable media comprising computer-executable instructions that, when executed, direct an application server to:

generate replies in response to client requests through use of a tag library containing pre-constructed tags for a variety of data formats, the client requests being submitted by diverse client devices that support different data formats and different communication protocols; and

structure the replies to define how the replies will appear when communicated to and presented on the client devices and independently form individual replies for output capabilities of the client devices so that the replies are encoded to comply with the data formats supported by the client devices and are sent using the communication protocols of the client devices.

34 One or more computer-readable media as recited in claim 33, further comprising computer-executable instructions that, when executed, direct an application server to use pre-constructed tags to structure the replies.

EVIDENCE APPENDIX

None.

RELATED PROCEEDINGS APPENDIX

None.